

ETL Proof of Concept Written Response: Pentaho Data Integration

Course: Evaluating ETL Tools and Technologies, afternoon session
ETL Vendors in Action

Table of Contents

Proof of Concept Overview	2
Scenario Overview	2
Demo Scenarios / Topics	4
1. View of ETL / Data Integration and Product Architecture Overview	4
2. Extract Scenario 1: Customer Dimension Incremental Extract	7
3. Extract Scenario 2: Shipments Fact Table Extract	8
4. Extraction Scenario 3: Open Case	9
5. Extraction Scenario 4: Time Dimension.....	11
6. Maintenance Features	9
7. Operations and Deployment	10
8. Pricing	11
9. Performance Features.....	14

Proof of Concept Overview

The scenarios for the proof of concept are all based on a wholesale business that supplies specialty products to retailers. The scenarios are based on the items that one might consider important when evaluating an ETL solution for a single data warehouse.

The examples are all built around a wholesale shipments schema, with a set of source tables loaded with data and a set of target tables to be populated by the tools. The extract rules for the schema are simple, but should be enough to demonstrate basic and some advanced capabilities in the products.

The afternoon will be a mix of discussion and demo, with the emphasis on showing how the products are used to accomplish specific tasks. While the focus is on extraction, some of the scenarios or presentation topics involve showing other features like metadata management, data profiling or monitoring job execution.

Because there's no way to show the entire set of ETL for three vendors in the time allotted, we'll be using different elements to show different features. For the scenarios listed we expect to see the features used to accomplish the task live. It isn't expected that we can see the entire extract constructed for each scenario in the time given. However, a complete set of extracts is required in order to show how dependencies, scheduling and monitoring work.

Demo time is limited so there are topics/scenarios labeled "time permitted" which we may not be able to show. They are included in case we have extra time at the end of the class.

Scenario Overview

In a proof of concept you provide to vendors all the source and target table definitions, extract rules and source data. Since this is meant to reflect the real ETL you'll be doing, it's a good idea to select both simple extracts and complex extracts or extracts that have problem data. When you provide this information, it should be formally documented so the vendor understands in detail what they are supposed to show.

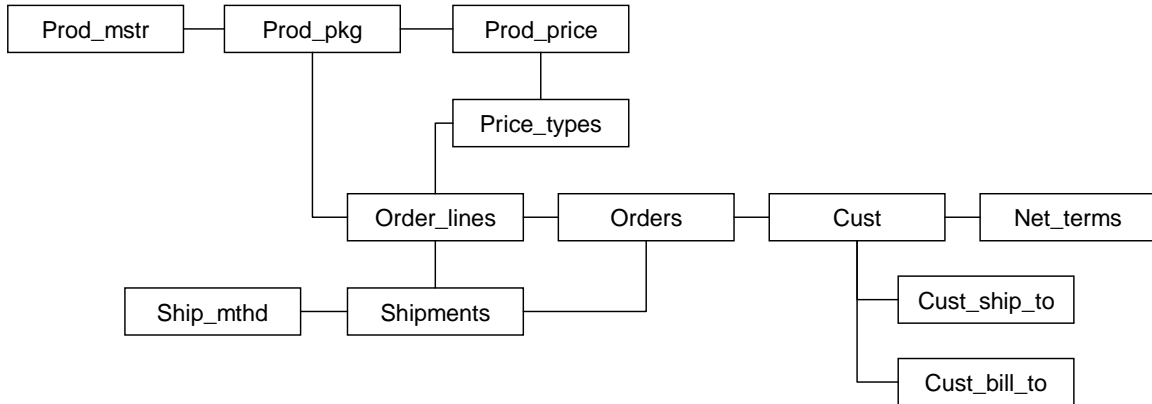
Part of the reason for selecting source data with quality problems is that this will show how a developer is expected to work within the tool. If all the extracts are based on ideal tables and data, as with standard vendor demos, then you won't see what a developer really has to face when dealing with data exceptions.

As a rule, you should have imperfect data, tables with relationship problems like different types on join or lookup columns, and you should always require the use of relational database in the proof of concept.

Using a database is important because it will show you how the tool interacts with a database. Many vendor demos you see are based on text files as input and output, which

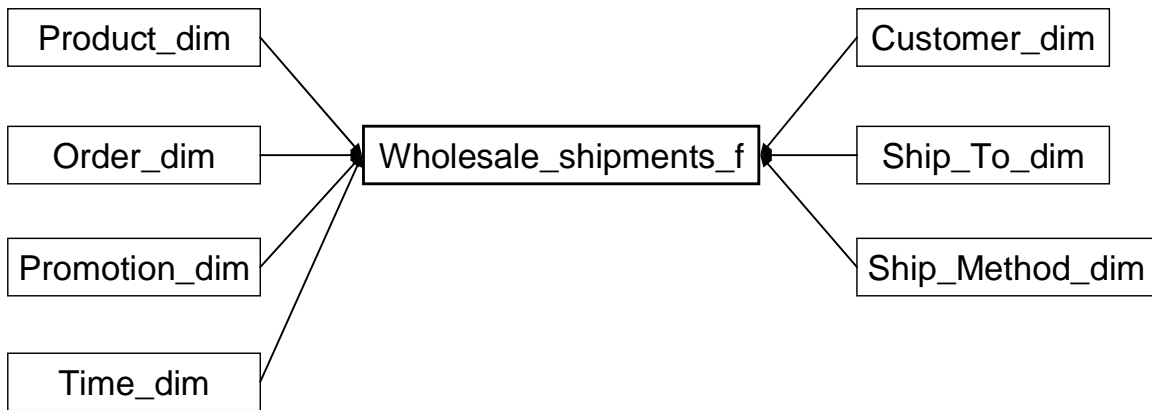
can avoid some of the difficulties when dealing with SQL since all the work is done directly in the ETL engine.

For our scenarios, we will be using the following source and target schemas. The source schema consists of 12 tables with some of the common design problems found in source databases.



Among the problems are a mix of second and third normal form, multi-part keys, invalid foreign key constraints, and multiple join paths to the same data elements. In addition to the above tables there are two change-data-capture tables for the shipment and customer data. These are used for incremental extract examples.

The target schema has seven dimension tables and one fact table. The fact table contains all shipments of order lines by day to customer addresses in the ship_to dimension.



There are several things in this target schema that complicate the extracts. The time dimension is based on a fiscal calendar using 4 and 5 week periods rather than a simple date-based dimension. There is no source for this data, so it must be constructed by the ETL job. A non-data-driven extract is a challenge for some ETL products. The ship_method dimension has unique rows based on a multi-part key which can cause trouble for some ETL tools' lookup functions. The specific details about extract rules and data are available at the end of this document.

Demo Scenarios / Topics

The following section describes each of the presentation topics or scenarios that will be reviewed during the course. For each scenario, there is a list of directions or questions to be answered and a short view of the schema and data (if applicable).

Included with the descriptions are sample criteria you might use during an evaluation so that you can score the vendors during the class. After this section there are responses from each of the vendors to all of the scenario questions so you have something to refer back to.

1. Product Architecture Overview and ETL Perspective

This is a short (4 minutes or less) presentation about the company, its products and its view of the ETL and data integration market. You should expect to hear answers to the following types of questions:

- The product name or components used for this demo, whether all components are included in the base product and / or how it's packaged.
 - a. Pentaho Data Integration (a.k.a. Kettle, a.k.a. PDI) components:
 - i. **Spoon:** Graphical design environment for creating and executing ETL transformations and jobs. This is the primary component used in the demonstration.
 - ii. **Pan:** Command-line execution of single, pre-defined transformations jobs
 - iii. **Kitchen:** Scheduler for multi-stage jobs
 - iv. **Carte:** Remote execution "slave" server with embedded web server
 - v. **Pentaho Data Integration Enterprise Console:** Thin-client administration environment for remote execution and monitoring of transformation and job performance, with the ability to set thresholds and receive alerts when min/max thresholds are exceeded.
 - vi. **Pentaho BI Platform:** Integrated scheduling of transformations or jobs; ability to call real-time transformations and use output in reports and dashboards
- What is the core ETL product offering?
 - i. The core open source offering includes all of the above components except the Enterprise Console. Spoon is the primary component of the offering.
- What optional components are provided?
 - i. Pentaho Data Integration Enterprise Edition comes with enhanced functionality (Enterprise Console), professional technical support, product expertise, certified software, software maintenance, and software assurance.
 - ii. Additional community and 3rd-party contributed plug-ins and connectors are available [here](#).

- How are components / editions bundled?
 - i. Community Edition: Includes all of the components with the exception of the Enterprise Console
 - ii. Enterprise Edition: Includes everything in the Community Edition as well as the Enterprise Console
- What requirements are there for server installs, client installs, etc.?
 - i. The minimum system requirements to run a transformation or job are very low. You need a system that supports the Java 1.4 runtime environment (or a higher version) – PDI runs on the Sun JVM, the one from IBM, Gnu, Classpath, Kaffe, etc.
 - ii. The system typically needs around 128MB of RAM of which around 25MB is the minimal used by the engine. If you do aggressive caching or in-memory lookups, you might need more memory.
 - iii. For Spoon (the graphical environment), the same requirements are in place, but you also need SWT support. That limits the out-of-the-box support to the following platforms:
 1. - Windows(95-Vista)
 2. - OSX (PPC/Intel)
 3. - Linux(Gtk-PPC/x86/x86_64)
 4. - AIX(Motif)
 5. - HPUX(Motif)
 6. - Solaris(Motif)
 - iv. Other than those requirements (JVM, OS), all software that is needed to run Pentaho Data Integration is included in the downloads we provide.
- What external dependencies are there? For example, a license for a database to be used as a metadata repository.
 - i. None. PDI uses either file-based or RDBMS-based storage of transformations and jobs. If you use RDBMS-based repository, you would need a license for that.
- What do you consider your sweet spot in applications or in industries?
 - i. The sweet-spot application for Pentaho Data Integration is ETL for data warehousing. We have customers who use it for data archival as well as data migration and even some EII, but the bulk of our customer success and our ongoing investments are centered around data warehousing use-cases.
 - ii. From an industry perspective, we have an inherently “pull-based” model where customers tend to find us, download our software, and come to us for additional functionality, services, indemnification, and more. We don’t have a specific industry focus. We have seen brisk adoption in Federal Government, Financial Services, Healthcare, Travel and Transportation, Web 2.0, and other industries. Any industry that has organizations that are sensitive to the costs of ETL and software in general tends to be fairly open to commercial open source software.

- What are your product's strongest points?
 - i. Ease of Use
 - 1. Graphical, drag-and-drop design environment with built-in validation, data previews, SQL previews, out-of-box support for Slowly Changing Dimensions, and an integrated ETL debugger
 - 2. Most users familiar with ETL concepts or other ETL tools like Oracle Warehouse Builder or Informatica typically pick up Pentaho Data Integration very quickly and many comment in our announcements and on our forums that they found it very easy to use.
 - ii. Performance and Scalability
 - 1. Pentaho Data Integration supports clustered deployment and massively parallel ETL.
 - 2. Pentaho Data Integration supports many of the bulk-loading interfaces of popular traditional and open source databases
- Why would a customer choose your product over others?
 - i. Over traditional proprietary ETL
 - 1. Cost – Pentaho Data Integration is significantly less expensive in up-front and ongoing costs compared to traditional tools like Informatica or Ab Initio, without any up-front software license fees.
 - 2. Flexibility – Pentaho Data Integration is flexible and extensible, and users are easily able to create their own optimized connectors to different sources.
 - 3. Performance and Scalability – Customers have replaced existing tools including IBM/Ascential and Ab Initio based on performance during a customer-run benchmark on their hardware with their data. Pentaho Data Integration was as-fast or faster than their incumbent tools in their benchmarks.
 - ii. Over other open source ETL
 - 1. Breadth – Pentaho provides a complete suite of Business Intelligence tools as well as data integration.
 - 2. Architecture – Pentaho Data Integration uses a metadata-driven architecture and avoids the scalability limitations and “black box” challenges (troubleshooting, tuning) of code-generation approaches.
 - 3. Proof – Pentaho Data Integration has numerous public references at companies of all sizes across many different industries, with real-world customers achieving peak data loading performance of 300,000 rows/second. No open source ETL vendor has the breadth and depth of real-world customer proof points that Pentaho Data Integration has.
 - 1. License – Pentaho Data Integration is provided with a non-viral Lesser General Public License license rather than a General Public License. Some organizations have concerns about adhering to the requirements of the GPL license. www.opensource.org has more information on open source license types, requirements, and implications.

2. Extract Scenario 1: Customer Dimension Incremental Extract

This scenario demonstrates the extract logic and facilities for a slowly changing dimension where history is preserved in the rows using start and end dates.

Scenario

Change rows are stored in a change data capture table. This is a table that mirrors the source table, but has additional columns that indicate whether a row is an insert, update or delete, and a timestamp for the activity. This is what you might see as the result of change replication using database replication facilities. The expectation is that the change table provides the activity, while the source table shows the current state of the data.

To properly test the extract, there are four rows for three different conditions. The test cases are as follows:

- Insert of a new customer
- Update of an existing customer, changing address line 2.
- Same day insert and later update of a row (within the same batch).

The goal of this scenario is to show how one would address the issues associated with slowly-changing dimension tables where a changed-data capture table is used as input. The intent is to also show some of the features available for looking at data values before and after update from within the developer's interface. The demonstration should answer the types of questions outlined below.

- How do you perform a simple lookup from another table?
 - i. [There are built-in steps for performing lookups.](#)
- Are there any features that automate or simplify the maintenance of slowly changing dimensions, or must you do this work yourself?
 - i. [Yes, there is a purpose-built "Dimension Lookup/Update" step to handle slowly changing dimensions.](#)
- How do you deal with both inserts and updates against a target table without requiring separate logic?
 - i. [Via a purpose-built Insert/Update step.](#)
- Do you have any problems with locking or staging when retrieving and updating rows in the target table as part of the extract?
 - i. [No. PDI is an ETL engine as opposed to a code generator, so it is not confined to the limitations of the database. Each applicable step in PDI creates a separate connection to the database, and in the event that you do encounter locking or you only want to use a single connection, the transformation can be configured to use unique connections.](#)
- How do you preview source data from within the design interface?
 - i. [Each data input step contains a "Preview" option to view the source data.](#)
 - ii. [There is also a "View" mode that will let you independently explore RDBMS content.](#)

- Can you look at both the input and output data from within the developer interface to see how updates were applied?
 - i. Yes.
- In the case where you do not preserve history, what steps do you have to take for a destructive load?
 - i. PDI transformations can be setup to automatically rollback on failure.
- Is there any support for managing hierarchies in dimensions?
 - i. No.

3. Extract Scenario 2: Shipments Fact Table Extract

The purpose of this scenario is to show a more complex extract involving conditional logic, calculations, many source tables, multiple lookups, and available facilities for dealing with surrogate key matching and validation for dimensional fact tables.

Scenario

The goal of this scenario is to show how one would address the more complex issues associated with building a fact table. Added complications in the data include common problems faced by developers. This includes dealing with missing data (forcing outer joins or conditional lookups), varchar and char data type conversions and comparisons, and missing values. The demonstration should provide answers to the following types of questions.

- How do you do a lookup based on a multi-column input key? (ship_method_key uses both ship_cd and carrier)
 - i. The Database Lookup step lets you specify multiple keys. An alternate solution would be to use the Database Join step, which is more generic and allows you to type in your own SQL.
- How would you accomplish the equivalent of an outer join between two tables? (the shipments table has some bad product values that won't join to the product tables, there is a non-existent customer number, and the ship_to address is missing for a shipment)
 - i. The Database Join step has an option to specify it as an outer join. When using the Database Lookup step, outer join is the default behavior, and there is an option to "do not pass the row if the lookup fails" which would perform an inner join.
- Assume the developer forgot to include a table in the metadata so it isn't present but is needed at design time. How does the developer import the metadata so the table can be used in this extract?
 - i. PDI will automatically generate the SQL necessary to create the table and allow you to execute it.
- How do you do calculations involving values from multiple tables, or aggregation? (unit calculations are an example, and missing rows in one table can cause difficulties with these calculations)

- i. PDI's streaming based engine lets you pull information from multiple tables, merge the streams, and then use a Calculation Step. For aggregations, there is a purpose-built Group By step.
- What features does the tool have to automate common tasks like keying fact tables?
 - i. One option would be use the Add Sequence step, which can use the database to get the sequence (i.e. Oracle) or use built-in generic counter.
 - ii. Another option would be to include it in the SQL when generating the fact table (i.e. AUTO_INCREMENT).
 - iii. The Table Output step has an option to return auto-generated key which could then be passed on downstream.

4. Extraction Scenario 3: Open Case

This is an open case. The vendors have been asked to demonstrate something that is unique or compelling about their products. The specific features they show aren't always known in advance.

5. Maintenance Features

Post-deployment maintenance is an important aspect of the ETL process. There is no specific scenario. Instead, the vendors have been asked to describe and demonstrate features available to developers that address the types of questions outlined below.

- Assume there is a change to the data type of the order_nbr column in the Orders dimension. What is the process of tracing the impact of this change and how would a developer change the affected extracts?
 - i. You can show the input values going into each step and find what step the change is coming from. In this example, if the data type of a field changes PDI will detect those changes give you the option to generate the SQL necessary to update/correct the table.
- What source control or version control features are available?
 - i. No native version control at this time, but it is on the roadmap. Currently we recommend using 3rd-party version control software (i.e. Subversion) and check PDI transformations/jobs (XML files) into that.
- What facilities are available for documenting or annotating extracts?
 - i. Each step in a transformation or job can be renamed to make the entire extract easily readable. Separate "Note" objects can also be placed on the design environment for additional annotation.
- How does a developer compare and find differences between an extract that has been deployed in production and one in the repository / development environment?
 - i. With a version control system like subversion you can see who changed what, when and why.
- How does a developer address production deployment and rollback of jobs?

- i. Using a standard version control system – checkout files from one environment to another, make copies, version them, create branches, etc.

6. Operations and Deployment

To show features available for scheduling and monitoring, we are using the complete set of extracts for the dimension and fact tables. The vendors have been asked to show how dependencies between the various extracts are configured, how a schedule for execution is created, and how the extract jobs are monitored. The goal is to show product features related to the types of questions outlined below.

- What is the executable unit of work in the product?
 - i. **Transformations** (.ktr files) and **Jobs** (.kjb files)
 1. Transformations move and transform rows from source to target.
 2. Jobs provide high level flow control of transformations (e.g. execute transformation1 > on success execute transformation 2 > email success/failure when complete).
- Demonstrate how execution dependencies between jobs are set up (e.g. fact table doesn't load until dimension tables have loaded).
- How do you make a change to the dependencies?
 - i. Using the graphical design environment (Spoon).
- How do schedule-based initiation and event-based initiation of jobs work?
 - i. Through integration with Pentaho's BI platform, PDI transformations/jobs can use its built-in scheduler. PDI jobs can also be created to wait for certain events to occur, like the arrival of a file on an FTP server for example.
 - ii. PDI can also integrate with existing 3rd-party enterprise schedulers via command line interface (Pan and Kitchen).
- How can execution be monitored by a developer within the development environment, and by an administrator outside the development environment?
 - i. Detailed step logging is available inside of the development environment.
 - ii. Pentaho Data Integration Enterprise Console provides remote monitoring, stop/start, and alerting functionality.
- Explain the mechanisms available for monitoring execution and sending alerts if there are problems.
 - i. Pentaho Data Integration Enterprise Console provides monitoring capabilities and allows you to define min/max thresholds on jobs/transformation and receive alerts when those thresholds are exceeded.
 - ii. Another efficient option is to have the data integration job send email on failure.

7. Extraction Scenario 4: Time Dimension

This scenario involves building a derived table where there is no data source. The job must construct a time dimension based on a fiscal (4-4-5) calendar with a number of derived date attributes. The vendors have been asked to explain and demonstrate how one would construct a derived table like this where there is no data source and the data must be created by the ETL program.

This sort of extract also shows how vendors deal with looping constructs and conditional logic, both of which are needed to work out problems like leap years and fiscal years crossing into new calendar years.

- What facilities are there to generate data?
 - i. Two built-in steps: Generate Rows step and Generate Random Data step
- Ease of addressing conditional logic and loops.
 - i. There are a number of built-in steps that could be used depending on the situation, including flow control steps like Abort and Blocking (wait until last row completes). The Filter Row step allows you to send data down different paths based on a certain criteria. There is also a Switch Case step.
- Date management and attribute functionality.
 - i. There are built-in date functions in the Calculator step, or you could use the JavaScript step, and there are a few things in the Formula step.
- Does the product recognize time as a specific type of dimension?
 - i. No.

8. Pricing

Since pricing is so variable we asked for information about the licensing policies and how costs are calculated.

For ETL , there is the added complication of sizing. Depending on the tools, you may need a large, small, or no ETL server. You may also need to expand the warehouse or source servers to accommodate the workload. How you configure your environment can affect the pricing of the software.

Basic criteria:

Is there a per seat cost for developers?

No.

Is there a per seat cost for administrators?

No.

Is there a price per server by CPU? Per core?

Yes, Pentaho Enterprise Editions are priced per CPU band (i.e. 0-4, 5-8, etc.). Dual-core CPUs are treated as one CPU. When running CPUs with more than 2 cores per socket, CPU count will be calculated by dividing the total cores by 2.

Is the price different for different server operating systems?

No.

Are there per source or per target server / instance charges?

No.

Are there additional source / target connector charges?

Some additional connectors are developed and sold by Pentaho Partners.

Is there a charge for development or test environments? If so, is it the same cost?

Pricing is per CPU. The CPU count includes development, test, and production environments whether virtualized or not.

How is maintenance charged? What is the range if it is a percent of some set of costs?

Pentaho Enterprise Editions (which includes software maintenance, technical support, and more) are charged on an annual subscription basis based on number of CPUs.

How many different editions or bundles are offered?

Two: the Community Edition and Enterprise Edition. The Community Edition is available for free download and use from www.pentaho.com. The Enterprise Edition is provided to Pentaho customers on an annual subscription basis and includes unlimited technical support, enhanced functionality, software maintenance, and more.

Are there additional charges for team-based development, e.g. source control, project-level security, role-based access?

No.

Please provide an estimated list price and support cost for these two scenarios:

Scenario 1: Department / project level ETL

A single data warehouse, with one ETL project

3 ETL developers, 1 administrator / operations role

2 different (non-legacy) database source types, and some file-based extracts

1 target database server, 4 CPUs

One production environment

One test and development environment

Small volumes of raw data moved through standard star-schema style batch extract, with the total target warehouse size of 180 GB of data (60GB of data loaded per year).

Standard next-business-day support

Note: please specify the number of servers and CPUs used to support this configuration

Estimated Total # of CPUs: 2 (1 production, 1 dev/test)

Price for single module subscription (i.e. Pentaho Data Integration Enterprise Edition) up to 4 CPUs: \$11,000 (annually)

Standard (a.k.a. Gold) support SLA: included

Total Price for scenario 1: \$11,000 (annually)

Scenario 2: Enterprise ETL

Multiple data warehouses/marts with several different ETL projects

10 ETL developers, 3 administrator / operations roles

3 different (non-legacy) database source types, file-based extracts, one SAP source system, requirement to consume XML formatted data

3 target database servers for warehouses / marts - 12 4, 4 CPUs respectively

3 production environments for the three projects (but infrastructure, repositories, etc. is centrally managed)

3 half-scale development environments

3 half-scale test environments

Moderate volumes of raw data moved through standard star-schema style batch extracts for two projects, with the total target sizes of each at 500 GB of data (~160 GB of data loaded per year).

One larger project environment with 2 TB of data (~650GB of data loaded per year), through more complex rules in batch mode, plus small amounts of data streaming in through your choice of either message queues / ESB / event publishing and then processed through your choice of either on-demand or in mini-batches. Specify the preferred option for streaming data consumption for the purpose of this pricing exercise. Assume an 8 CPU database server is this is important.

Note: this environment requires team-based development support and project-level security and roles.

Enterprise level (same-day response) support including off-hours and weekends

Note: please specify the number of servers and CPUs used to support this configuration

Total # of CPUs: 32 [16 production (8 for large env., 4 each for the other 2; 16 for dev and test env. (½ of each production env.))]

Price for single module subscription (i.e. Pentaho Data Integration Enterprise Edition) for 17 to 32 CPUs: \$69,000 (annually)

Upgrade to 24x7x365 support: \$3,300 (annually)

Upgrade support SLA from gold to platinum: \$3,300 (annually)

Total Price for scenario 2: \$75,600 (annually)

9. Performance Features

There is no easy way in a class setting to demonstrate performance. Instead, the vendors have been asked to describe features in the product that specifically address performance needs and answer common performance questions like the following.

- What features are available to deal with large volumes of source data?
 - i. Parallel reading for text files
 - ii. Setup multiple table input steps to read in parallel
 - iii. Clustered execution
 - iv. Bulk loaders and extract steps for Oracle and MySQL
- How does a developer set up or create an extract that can execute in parallel? Does it require special transforms or is it the same ETL logic regardless of parallel degree? Does it require changes to the database like table partitioning?
 - i. Same ETL logic, but clustered environment requires some additional setup. Table partitioning is transparent to PDI.
- What features are available for caching of source data or lookup values, and what are the limitations?
 - i. Lookup step can enable caching
 - ii. Stream Lookup step only reads from memory
 - iii. Only limitation is the amount of memory
- How can the product be configured to run extracts on more than one physical server, if this capability is available? What must the developer do differently to take advantage of these features?
 - i. Yes, you can setup a clustered database connection in PDI
- Can individual extracts be parallelized to run across servers?
 - i. Yes, PDI can be clustered using the Carte server. ETL logic remains the same, you just specify during execution to execute it across the cluster.
- Are there grid / concurrent processing capabilities? If so, how do they work?
 - i. Grid/Cloud computing is possible in version 3.2 with the introduction of dynamic slaver server registration, for situations where hosts are being added or removed at will. Slave servers can be configured to talk to one or more master servers with build in failover and load balancing on the roadmap.